IMAGE PROCESSING

The present invention is concerned with image processing.

5   According to the present invention there is provided a method of processing a digitally coded image in which picture elements are each represented by a colour value, comprising, for each of a plurality of said picture elements:

(a) performing a plurality of comparisons, each comparison comprising comparing a first picture element group which comprises the picture element under consideration and at least one
10   further picture element in the vicinity thereof with a second picture element group which comprises a base picture element and at least one further picture element, the number of picture elements in the second group being the same as the number of picture elements in the first group and the position of the or each further element of the second group relative to the base picture element of the second group being the same as the position of the or a respective further element of the first
15   group relative to the picture element under consideration, wherein each comparison determines whether the two groups match in the sense that they meet a criterion of similarity; and

(b) when at least one comparison results in a match, computing a replacement colour value for the picture element under consideration, the replacement colour value being a function of the colour value for the base picture element of the or each second group for which a match was
20   obtained.

Preferably, the method includes identifying picture elements which meet a criterion of distinctiveness, and computing a replacement colour value only for picture elements not meeting the distinctiveness criterion.

Other, preferred, aspects of the invention are defined in the claims.

25   Some embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of an apparatus for performing the invention;

Figure 2 is a flowchart of the steps to be performed by the apparatus of Figure 1 in accordance with one embodiment of the invention;

30   Figure 3 is a similar flowchart for a second embodiment of the invention;

Figure 4 is a similar flowchart for a third embodiment of the invention; and

Figures 5 to 8 illustrate the effects of this processing on some sample images.

Figure 1 shows an apparatus consisting of a general purpose computer programmed to perform image analysis according to a first embodiment of the invention. It has a bus 1, to which

2

are connected a central processing unit 2, a visual display 3, a keyboard 4, a scanner 5 (or other input device, not shown) for input of images, and a memory 6.

In the memory 6 are stored an operating system 601, a program 602 for performing the image analysis, and storage areas 603, 604 for storing an image to be processed and a processed

5   image, respectively. Each image is stored as a two-dimensional array of values, each value representing the brightness and/or colour of a picture element within the array.

In a first embodiment of the invention, the program 602 is arranged to operate as shown in the flowchart of Figure 2. The image to be processed is stored as an array [C] of pixels where the position of a pixel is expressed in Cartesian co-ordinates e.g. (x_1, x_2) or as a vector (in bold

10   type) e.g. $x = (x_1, x_2)$. The colour of a pixel at x is stored as a vector C(x) consisting of three components. In these examples r,g,b components are used but other colour spaces could be employed. In a monochrome system C would have only one (luminance) component. The results of this process are to be stored in a similar array C_{OUT}.

The process is iterative and starts at Step 100 which simply indicates that one begins with

15   one pixel x and repeats for each other pixel (the order which this is performed is not significant), exiting the loop at Step 102 when all have been processed. However it is not essential to process all pixels: some may be deliberately excluded, for reasons that will be discussed presently.

In Step 104, a comparison count I is set to zero, a match count M is set to 1, and a colour vector V is set to the colour at x. V has three components which take values according to the

20   colour space employed e.g. (r,g,b).

Step 106: n (typically 3) random pixels at $x'_i = (x'_{i1}, x'_{i2})$ $i = 1, \ldots n$ are selected in the neighbourhood of x where

$| x'_{ij} - x_j | < r_j$ for all $j = 1,2$ and r_j defines the size of a rectangular neighbourhood (or square neighbourhood with $r_1 = r_2 = r$). A typical value for r_j would be 2 for a 640 x 416 image.

25   Step 108: A pixel at $y = (y_1, y_2)$ is then randomly selected elsewhere in the image and (Step 110) the comparison count I incremented. This pixel is selected to be $\geq r_j$ from the image boundary to avoid edge effects. If desired, the choice of y could be limited to lie within a certain maximum distance from x. If, at Step 112, the value of I does not exceed the value of a threshold L (typical values are 10 – 100) a test for a match between the neighbourhoods of x and y is carried

30   out.

Step 114: Let the colour of the pixel at x be $C(x) = (C_1(x), C_2(x), C_3(x)) = (r_x, g_x, b_x)$

Then the neighbourhoods match if each of the pixels x, x'_i (that is, the pixel under consideration and its n neighbouring pixels) matches the corresponding pixel at y, y'_i, where the positions of y'_i relative to y are the same as those of x'_i relative to x. That is to say:

35          $x - x'_i = y - y'_i$ for all $i = 1, \ldots, n$.

where pixels at x and y are deemed to match if

$|C_j(x) - C_j(y)| < d_j$ for all $j = 1, 2, 3$.

and similarly for x'$_i$ and y'$_i$:

$|C_j(x'_i) - C_j(y'_i)| < d_j$ for all $j = 1,2,3$ and all $i = 1, ..., n$.

where d_j is a threshold that determines whether colour component j is sufficiently different to constitute a pixel mismatch. In the tests described below, the colour components were represented on a scale of 0 to 255 and a single value of $d_j = 80$ was used. In general d_j may be dependent upon x. For example, it may be preferred to model attention so that less emphasis is given to darker regions by increasing d_j in these areas.

If a match is found then at Step 116 the counter M is incremented and the values of the colour components at y are added to V.

$V = V + C(y)$

Following a match the process returns to Step 106 of selecting a fresh neighbourhood around x containing n random pixels, whereas if no match is found it returns to Step 108 to select a new y without changing the pixel neighbourhood.

If at Step 112 the value of I exceeds the threshold L, the colour of the pixel at $x = (x_1,x_2)$ in the transformed image is given (Step 118) the average value of the colours of the M pixels found to have matching neighbourhoods i.e.

$C_{OUT}(x) = V / M$.

This process is repeated from Step 100 until all pixels in the image have been dealt with. The resulting transformed image possesses a much reduced spread of colours but also contains small levels of noise arising from the random nature of the algorithm. This noise can be simply removed (Step 120) by applying a standard smoothing algorithm. In this embodiment a pixel is assigned the average colour of the pixels in the surrounding 3 x 3 window.

The algorithm shown in Figure 2 processes all pixels x, and all will have their colours altered except in the case of pixels whose neighbourhoods are so dissimilar to the rest of the image that no matches are found. In that the process necessarily involves a loss of information, we prefer to identify important parts of the image and exclude these. Thus the embodiment of Figure 3 excludes regions of interest from the filtering process. In Figure 3, those steps which are identical to those of Figure 2 are given the same reference numerals.

The process begins at Step 130 with the generation of a saliency map consisting of an array of attention scores $Scores(x_1,x_2)$ using the method described in our international patent application WO01/61648 (also published as US 20020080133). Other methods of generating saliency maps may also be used although their performance may not always be best suited to this application: See for example L Itti, C Koch and E Niebur, "A model of saliency-based visual

attention for rapid scene analysis," IEEE Trans on PAMI, vol 20, no 11, pp 1254-1259, Nov 1998; W Osberger and A J Maeder, "Automatic identification of perceptually important regions in an image," Proc 14th IEEE Int. Conf on Pattern Recognition, pp 701-704, August 1998; and W Osberger, US Patent Application no 2002/0126891, "Visual Attention Model," Sept 12 2002. The

5   values of $Scores(x_1,x_2)$ assigned to each pixel at $\mathbf{x} = (x_1,x_2)$ or a subset of pixels in an image reflect the level of attention at that location.

A value is given to the variable T, typically 0.9, which sets a threshold on $Scores(x_1,x_2)$ and determines whether the colour of the pixel at x is to be transformed or not where

$$Threshold = T* (max - min) + min$$

10   and $max = \underset{x_1,x_2}{Max} (Scores(x_1,x_2))$ $min = \underset{x_1,x_2}{Min}(Scores(x_1,x_2))$. However, other means of

calculating the value of *Threshold* may be used some of which can be dependent upon x.

If at Step 132 the value of $Scores(x_1,x_2)$ is greater than Threshold, the pixel in the original image at x is, at Step 134, copied unchanged into the transformed image array $C_{OUT} (x_1,x_2)$. This pixel represents a point of high attention in the image and will not be altered by this process.

15   The remainder of the process is as previously described: note however that, owing to the test at 132, in the smoothing algorithm, the colour value is replaced by the smoothed value only for those pixels whose attention scores are less than the value of *Threshold*.

Another embodiment is shown in Figure 4 in which attention scores are not computed beforehand. Instead, when at Step 112 the comparison count I exceeds the threshold L, a test is

20   performed at Step 150 to determine whether the match count M is greater than a threshold mt. If so, then, as before, the colour of the pixel at (x_1,x_2) in the transformed image is given, at Step 118, the average value of the colours of the M pixels found to have matching neighbourhoods i.e. V / M.

If, however, M is less than or equal to mt, the pixel in the original image at x is, at Step 152, copied unchanged into the transformed image array $C_{OUT}(x_1,x_2)$. This means that pixels

25   representing areas of high attention will be unlikely to be altered because only low values of M will be obtained in these image regions.

The degree of filtering that is applied to the image may be controlled by selecting the value of the thresholds d_j. Alternatively, or in addition, the filtering process can if desired be repeated: as shown at Step 170 in all three versions. The transformed image may be reloaded

30   whilst (in the case of Figure 4) retaining the original attention scores $Scores(x_1,x_2)$ and the whole process repeated to obtain successive transformations and greater suppression of background information.

Note that where random selection is called for, pseudo-random selection may be used instead.

5

Once filtering is complete, the transformed image may if desired be encoded in JPEG format (or any other compression algorithm) as shown at Step 180. The reduction in information contained in regions of low interest enables higher compression performances to be attained than on the original image.

5   The results of applying the algorithm of Figure 4 to a football source are shown in Figure 5, which shows, from left to right, the original image (GIF format), the image after JPEG coding, and the image after filtering followed by JPEG CODING. Histograms of the distribution of hue values in the range 1-100 are also shown. It is found that the filtering reduces the compressed image file size from 13719 bytes to 10853 bytes.

10   Typically two iterations of pixel colour replacement and smoothing are sufficient, but this can be extended depending upon the colour reduction required.

Figure 6 illustrates how background information may be substantially removed whilst preserving important features of the image such as the boat and the mountain outline. The original JPEG encoding (Figure 6a) occupies 13361 bytes which is reduced to 10158 bytes after processing

15   once and JPEG encoding the transformed version (Figure 6b). The output image is reprocessed using the same VA scores and obtains a file size of 8881 bytes (Figure 6c). A further iteration obtains a size of 8317 bytes (Figure 6d).

This method may be applied with advantage to images containing artefacts (such as JPEG blocking effects). The re-assignment of colours to background regions tends to remove artefacts

20   which normally possess some similarity to their surroundings (See Figure 7, where the original image is shown on the left: on the right is shown the image obtained following processing with this method and subsequent re-coding using JPEG). However, artefacts which are very obtrusive and interfere with the main subject material will not be removed.

A further application of the method is the enhancement of figure-ground or the removal

25   of background distractions for improved recognisability. This application is illustrated in Figure 8 in which the background is almost completely replaced with a constant colour and the image of the dog is much more prominent. The method could therefore be applied to the processing of images displayed in a digital viewfinder in a camera where the enhancement of subject material will assist photographers to compose their pictures.

30   Essential visual information is retained in the transformed images whilst reducing the variability of colours in unimportant areas. The transformed image thereby become much easier to segment using conventional algorithms because there are fewer colour boundaries to negotiate and shape outlines are more distinct. This means that this method will enhance the performance of many conventional algorithms that seek to partition images into separate and meaningful

35   homogeneous regions for whatever purpose.

In the embodiments we have described, the replacement colour value used is the average of the original value and those of all the pixels which it was found to match (although in fact it is not essential that the original value be included). Although it practice this does not necessarily result in a reduction in the number of different colour values in the image, nevertheless it results in

5   a reduction in the colour variability and hence – as has been demonstrated – increases the scope for compression and/or reduces the perception of artefacts in the image. Other replacement strategies may be adopted instead. For example, having obtained the average, the replacement could be chosen to be that one of a more limited (i.e. more coarsely quantised) range of colours to which the average is closest. Or the match results could be used to identify groups of pixels which could then

10   all be assigned the same colour value.

These embodiments assume that low level segmentation algorithms should not be applied to those areas in an image that merit high visual attention. Such regions are naturally anomalous and contain a high density of meaningful information for an observer. This means that any attempt to segment these areas is likely to be arbitrary because there is little or no information in the

15   surrounding regions or elsewhere in the image that can be usefully extrapolated. On the other hand less significant parts of the image that are more extensive can justifiably be transformed using quite primitive and low level algorithms. Paradoxically, distinctive object edges in an image attract high attention and therefore are not subjected to alteration in this approach. In fact the edges of objects at the pixel level in real images are extremely complex and diverse and would need specifically

20   tailored algorithms to be sure of a correct result in each case.

The second and third embodiments of the invention offer an approach to colour compression that makes use of a visual attention algorithm to determine visually important areas in the image which are not to be transformed. This approach therefore possesses the significant advantage that the process of assigning region identities does not have to address the difficult problem of defining

25   edges which normally hold the highest density of meaningful information. Non-attentive regions are transformed according to parameters derived from the same VA algorithm which indicates those regions sharing properties with many other parts of the image. The visual attention algorithm does not rely upon the pre-selection of features and hence has application to a greater range of images than standard feature based methods which tend to be tailored to work on categories of

30   images most suited to the selected feature measurements. Pixels in the regions subject to transformation are assigned an average colour and increased compression obtained through JPEG encoding or any other compression standard. Compression is applied to the least attentive regions of the image and therefore is unlikely to affect the perceptual quality of the overall image. The algorithm may be iteratively applied to the transformed images to obtain further compression at the

35   expense of more background detail.